CS7646 - Project 8: Strategy Evaluation

Apurva Gandhi
agandhi301@gatech.edu

***Abstract*—**This report contains an analysis of a stock trading project, focusing on a manual strategy, a machine learning-based strategy learner, and a benchmark. The manual strategy combines five technical indicators, demonstrating superior performance in both in-sample and out-of-sample periods compared to the benchmark. The strategy learner, employing a BagLearner with an RTLearner, outperforms both the manual strategy and the benchmark. Additionally, an experiment investigates the impact parameter's influence on the strategy learner's performance, revealing a decrease in profitability with higher impact values.

INTRODUCTION

This project serves as the capstone for the ML4T course, integrating investing and machine learning concepts by implementing and comparing a manual trading strategy and a learning strategy. The manual strategy involves building a rule-based trader using a minimum of three selected indicators and comparing its performance against a benchmark. Simultaneously, the strategy learner utilizes one of the learners implemented during the course and learns trading strategies during the training phase, subsequently evaluated in the testing phase. Experiments focus on comparing in-sample and out-of-sample trading behavior, specifically on the symbol JPM for this report. The overarching goal is to deepen understanding of trading solutions and contribute to foundational knowledge in trading, machine learning, and investing. The report contains data and charts gathered by conducting experiments with different values with the manual learner, strategy learner, and benchmark. The hypothesis is that the manual learner will outperform the benchmark, and the strategy learner will outperform both the manual learner and the benchmark.

INDICATOR OVERVIEW

Implementation of Manual Learner and Strategy learner uses 5 indicators to make the appropriate trading decision. The indicators used in the Manual Strategy and Strategy Learner are Simple Moving Average, Momentum, Commodity

Channel Index, Bollinger Band Percentage, and Moving Average Convergence Divergence. A Simple Moving Average (SMA) is a calculation that averages a number of data points over a specific period. It is implemented by summing up the data points over a given window size and then dividing by the window size. The parameter that can be optimized is the window size. Momentum is a measure of the rate of change of a security's price. It is implemented by dividing the current price by the price a certain number of days ago and subtracting 1. The parameter that can be optimized is the window size. The Commodity Channel Index is a momentum-based oscillator used to help determine when an investment vehicle is reaching a condition of being overbought or oversold. It is implemented by calculating the CCI of the given values over a specified window size. It first calculates the rolling mean and mean deviation of the given values. The CCI is then calculated as the difference between the current value and the mean, divided by a multiple of the mean deviation. The parameter that can be optimized is the window size. This function calculates the Bollinger Band percentage. It first calculates the rolling mean and standard deviation of the given values over a specified window size. Then it calculates the upper and lower Bollinger Bands by adding and subtracting a multiple of the standard deviation from the mean. The Bollinger Band percentage is then calculated as the difference between the current value and the lower band, divided by the difference between the upper and lower bands. For both strategies, the Bollinger band is used to generate the Bollinger band percentage to reduce the implementation complexity of making the trading decision. For all 4 of the above indicators, both strategies do not optimize the window size parameter and use the standard set window size of 10, 10, 10, and 5. Moving Average Convergence Divergence is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. It calculates the MACD of the given values. It is implemented by first calculating the short-term and long-term exponential moving averages (EMAs) of the values. The MACD line is then calculated as the difference between the short-term and long-term EMAs. The signal line is calculated as the EMA of the MACD line over a specified signal period. The MACD histogram is then calculated as the difference between the MACD line and the signal line.

MANUAL STRATEGY

0.1 Strategy Description

The Manual Strategy combines five different technical indicators to generate an overall trading signal. These indicators are Simple Moving Average (SMA), Momentum, Commodity Channel Index (CCI), Bollinger Bands Percentage (BBP), and Moving Average Convergence Divergence (MACD). Each of these indicators generates a buy or sell signal based on certain conditions. For example, a buy signal is generated when the SMA is greater than the current price, and a sell signal is generated when the SMA is less than the current price. Similar conditions are applied for the other indicators. The individual signals from these indicators are then combined to create an overall signal. This is done by taking the mode (most frequent value) of the individual signals. This means that the overall signal will be a buy if the majority of the individual indicators suggest a buy, and a sell if the majority suggest a sell. The strategy then enters or exits positions based on this overall signal. If the signal is a buy and there are no current holdings, it buys 1000 shares. If there are already short holdings (-1000 shares), it buys 2000 shares to reverse the position. Similarly, if the signal is a sell and there are no current holdings, it sells 1000 shares. If there are already long holdings (1000 shares), it sells 2000 shares to reverse the position. This strategy is effective because it uses multiple technical indicators to generate trading signals. Each indicator provides a different perspective on the market, and by combining them, the strategy can potentially capture a more comprehensive view of the market. By combining these indicators, the strategy can generate a buy signal when the majority of the indicators suggest a bullish trend, and a sell signal when the majority suggest a bearish trend. This potentially increases the accuracy of the signals and improves the performance of the strategy.

0.2 Performance Analysis

As seen in Figure 1 and Figure 2, the performance of the Manual Strategy outperforms the benchmark for both in-sample and out-sample. Figure one shows how two different ways of investing $100,000 in the stock "JPM" from January 1, 2008, to December 31, 2009, compare to each other. The purple line is what happens if you buy 1000 shares on the first day and keep them until the end. The red line is what happens if you use a special strategy that tells you when to buy or sell. The black lines mean you sell your shares and the blue lines mean you buy them. The numbers on the y-axis show how much your money grows or shrinks over time. Both start at 1, which means $100,000. This is the data that we used to make the strategy, so we expect it to do well. The strategy does better than the purple line and makes about 80 percent more money by the end. To evaluate the

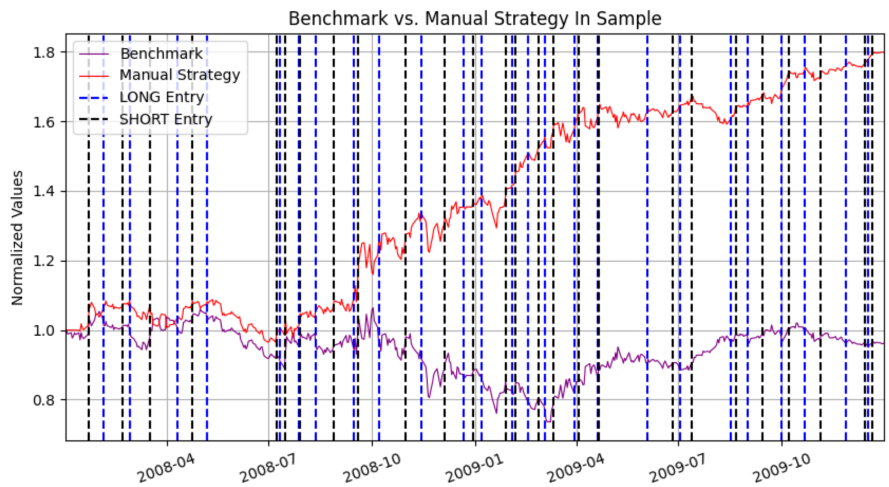

Figure 1—Benchmark vs. Manual Strategy In Sample

performance of the model on unseen data, we use Figure 2 to show the out-of-sample portfolio values. The settings are the same as in Figure 1, except for the time period, which is from January 1, 2010, to December 31, 2011. The red line indicates the manual strategy, which still outperforms the benchmark (purple line). However, the return is lower than in the in-sample period, with only a 13 to 15 percent better than the benchmark return. This is expected, as the model should perform better on the data it was trained on. It is important to note that the model was not adjusted by looking at the test data. The manual strategy has generalized well for this dataset and achieved good results on unseen data. As evidenced by Table 1, Manual Strategy outperforms benchmark, and Manual Strategy performs about 75 to 80 percent better compared to Benchmark during in-sample testing. Similarly, the Manual Strategy performs around 14 percent better compared to Benchmark during out-sample testing.

STRATEGY LEARNER

The trading problem was framed as a supervised learning problem. The learner was trained to predict future price movements based on historical price data and technical indicators. The technical indicators used were Simple Moving Average

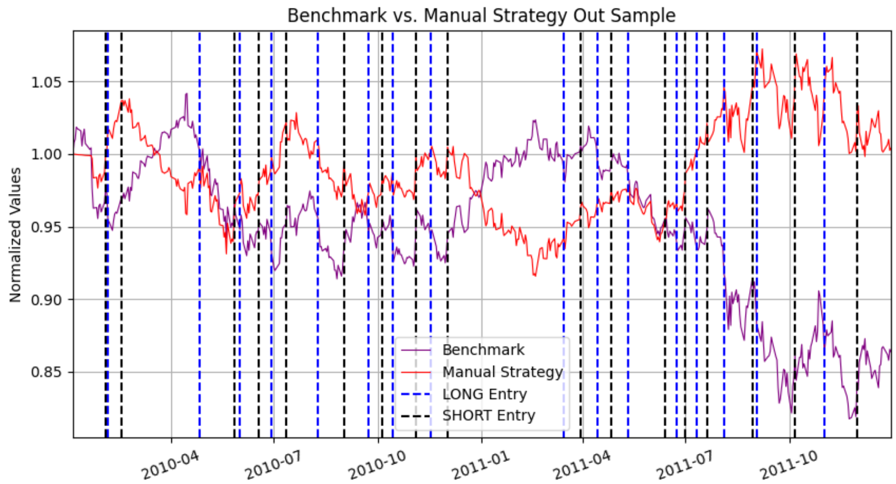

Figure 2—Benchmark vs. Manual Strategy Out Sample

	Cumulative Return	Standard deviation of daily returns	Mean of daily returns
Benchmark In-Sample	-0.037925	0.017468	0.000075
Benchmark Out-Sample	-0.136246	0.008961	-0.000251
Manual Strategy In-Sample	0.797222	0.012910	0.001247
Manual Strategy Out-Sample	0.004039	0.008012	0.000040

Table 1—Performance of Manual Strategy Vs. Benchmark

Ratio, Bollinger Band Percentage, Momentum, Commodity Channel Index, and Moving Average Convergence Divergence. The classification-based learner was used to create a strategy. The learner used was a BagLearner with an underlying RTLearner. The hyperparameters for the BagLearner were learner: RTLearner, kwargs: "leaf_size":5, bags: 20, boost: False, verbose: False. The leaf_size hyperparameter for the RTLearner was set to 5. This was determined empirically by testing different values and choosing the one that gave the best performance on the training data. The bags hyperparameter for the BagLearner was set to 20. This was also determined empirically by testing different values and choosing the one that gave the best performance on the training data. The data was not discretized or standardized. This is because the technical indicators used are already normalized to a certain extent. For example, the Bollinger Band Percentage is always between -1 and 1, and the Moving Average Convergence Divergence is a difference between two moving averages, so it does not have a specific range. The Momentum is also a ratio and does not need to be standardized. The Commodity Channel Index is typically between -100 and 100. Therefore, standardizing the data was not necessary. The learner was trained using the add_evidence method, which takes in a symbol, start date, end date, and starting value. It calculates the technical indicators for the given symbol and time frame, prepares the training data, and adds it to the learner. The learner was tested using the testPolicy method, which also takes in a symbol, start date, end date, and starting value. It calculates the technical indicators for the given symbol and time frame, queries the learner to get the predicted labels, and generates trades based on these predictions. As evident in Table 2, Strategy Learner has performed significantly better

	Cumulative Return	Standard deviation of daily returns	Mean of daily returns
Benchmark In-Sample	-0.037925	0.017468	0.000075
Benchmark Out-Sample	-0.136246	0.008961	-0.000251
Strategy Learner In-Sample	1.112905	0.008492	0.001521
Strategy Learner Out-Sample	0.346138	0.004587	0.000602

Table 2—Performance of the Strategy Learner Vs. Benchmark

than Benchmark during in-sample and out-sample. During in-sample testing, Strategy Learner has performed almost 115% better as compared to Benchmark. Similarly, Strategy Learner has performed around 47% better than Benchmark during out-sample testing. This suggests that the Strategy Learner is effective.

EXPERIMENT 1 (MANUAL STRATEGY / STRATEGY LEARNER)

The experiment is designed to compare the performance of a manual strategy, a strategy learner, and a benchmark strategy on a stock trading problem. The stock symbol used for the experiment is "JPM". The experiment is conducted both in-sample (from January 1, 2008, to December 31, 2009) and out-of-sample (from January 1, 2010, to December 31, 2011). The initial portfolio value is set to $100,000. The benchmark strategy simply buys 1000 shares of the stock at the beginning of the period and holds them until the end. The trades generated by each strategy are converted to a format suitable for the marketsim function using the helper function. The marketsim function computes the portfolio values for each strategy over time. The portfolio values are then normalized and plotted for comparison. Two plots are generated: one for the in-sample period and one for the out-of-sample period. Each plot shows the normalized portfolio values for the manual strategy, the strategy learner, and the benchmark strategy. The initial experimental hypothesis is that the strategy learner will outperform both the manual strategy and the benchmark strategy. This is based on the assumption that the strategy learner can effectively learn from the historical price data and technical indicators to generate profitable trades. Figure 3 shows that

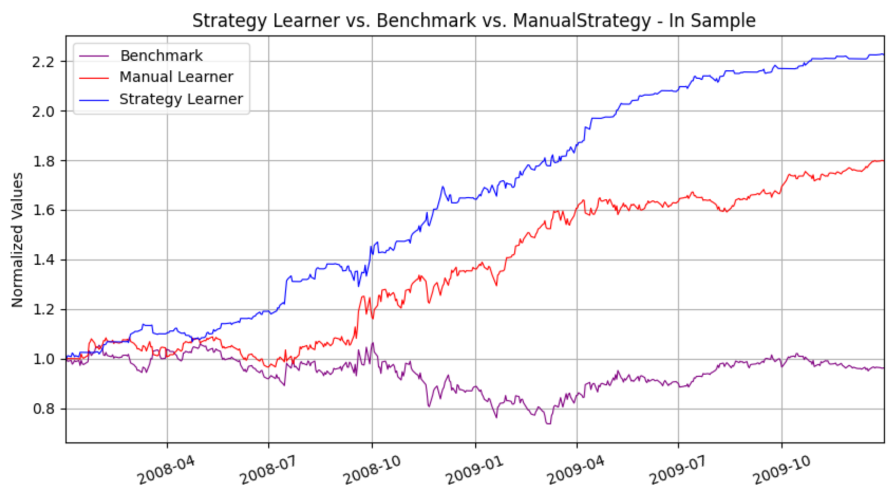

***Figure* 3**—Strategy Learner vs. Benchmark vs. ManualStrategy - In Sample

Strategy Learner performs better than Manual Learner and Benchmark in both in-sample and out-of-sample testing. In the in-sample period, Strategy Learner returns 120% more than Benchmark and 40% more than Manual Learner. Manual Learner also beats Benchmark by 80%. I expect these results to be consistent with the in-sample data because the model is trained on this data and therefore Strategy Learner should have an advantage. In the out-of-sample period, Strategy Learner still outperforms both Manual Learner and Benchmark, but with lower returns than in the in-sample period. Manual Learner does not improve much, but Strategy Learner continues to do well and returns almost 45% more than Manual Learner. Strategy Learner is able to generalize better and adapt to

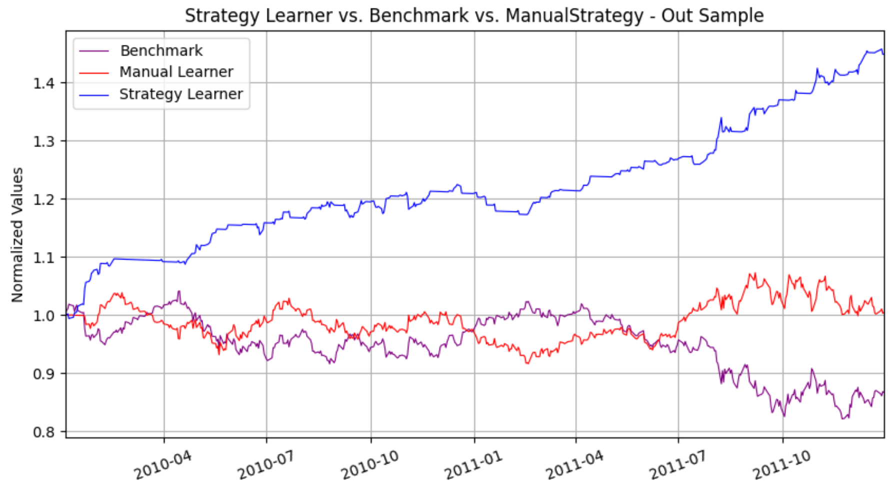

Figure 4—Strategy Learner vs. Benchmark vs. ManualStrategy - Out Sample

new data.

EXPERIMENT 2 (STRATEGY LEARNER)

The experiment examines the effect of the impact parameter on the performance of the strategy learner. The impact parameter represents the market impact of a trade, which is the change in price caused by the trade. Initial portfolio value was set to $100,000 and the symbol to "JPM", which is the stock of JPMorgan Chase & Co. The experiment trained and tested the strategy learner on the in-sample data, which was from January 1, 2008, to December 31, 2009. Furthermore, the experiment used three different values for the impact parameter: 0.005, 0.01, and 0.025. For each value, we calculated the total return, volatility, and average daily return of the portfolio. We then compared these metrics for each impact value to see how they affect the performance of the strategy learner. The hypothesis is that Changing the value of the impact parameter, which represents the impact of trades on the market price, should affect the in-sample trading behavior and results. Specifically, as the impact increases, we expect the total return to decrease and the volatility of the portfolio to increase. This is because a higher impact means that our trades have a larger effect on the market price, which could make it more difficult to execute profitable trades. Three metrics are used to evaluate the performance: total return, volatility, and average daily return. Total return is the percentage change in the portfolio value from the start to the end of the trading period. Volatility is the standard deviation of the daily returns, which indicates how much the portfolio value fluctuates. Average daily return is the mean of the daily returns, which indicates how much the portfolio value grows or shrinks on average. We used three different values for the impact parameter: 0.005, 0.01, and 0.025. Figure 5 compares the performance of the JPM stock for different impact values in the in-sample period. The impact values are 0.005 (purple), 0.01 (red), and 0.025 (black). The figure shows that the return decreases as the impact increases, which supports our hypothesis. This means that a higher impact reduces the profitability of the strategy. Table 3 shows the metrics for

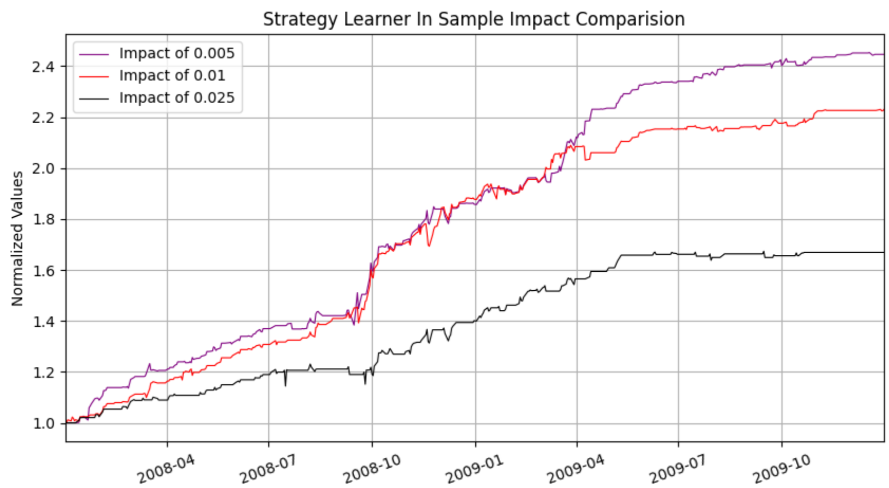

Figure 5—Strategy Learner In Sample Impact Comparison

	Cumulative Return	Standard deviation of daily returns	Mean of daily returns
Impact of 0.005	1.446396	0.007640	0.001805
Impact of 0.01	1.230286	0.007247	0.001619
Impact of 0.025	0.668915	0.007227	0.001043

Table 3—Summary of Experiment 2 - Strategy Learner with Different Impacts

the in-sample trading performance for different impact values. We can see that the cumulative return decreases as the impact increases. The cumulative return for an impact of 0.005 is 1.4464, for an impact of 0.01 is 1.2303, and for an impact of 0.025 is 0.66889. This confirms our hypothesis that a higher impact reduces the profitability of the strategy. The volatility, however, does not change much across different impact values. It might be because the impact difference is not significant enough to affect the volatility. The volatility difference is very small and does not contradict our hypothesis by a large margin. Overall, the graph and the table follow the trend of our hypothesis that increasing impact leads to decreasing return.